

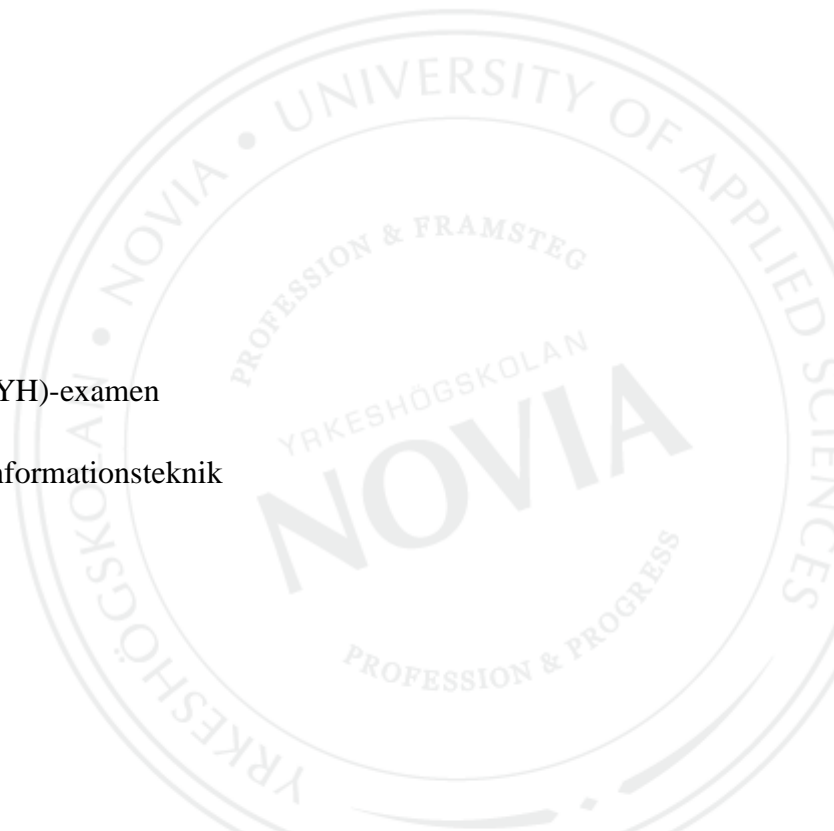
HJÄLPPROGRAM FÖR RÖNTGENSTUDERANDEN

Mikael Norrbäck

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för informationsteknik

Vasa 2012



EXAMENSARBETE

Författare: Mikael Norrbäck
Utbildningsprogram och ort: Informationsteknik, Vasa
Handledare: Kaj Wikman

Titel: *Hjälpprogram för röntgenstuderanden*

Datum 29.5.2012 Sidantal 24 Bilagor 0

Abstrakt

Examensarbetet utfördes som uppdrag åt utbildningsprogrammet för Radiografi och strålbehandling vid Yrkeshögskolan Novia. För att underlätta för utbildningsprogrammet önskades ett program som skulle kunna användas som hjälpmedel vid utlärn timer av nativbildstagning. Till arbetet hörde att planera och skapa ett program med hjälp av det material som arbetsgivaren ville ha med. Programmet är gjort i programmeringsspråket C# av projekttypen WPF och innehåller en virtuell röntgen, samt ett informationscenter för röntgenstuderanden.

Språk: svenska Nyckelord: virtuell röntgen, C#, WPF

BACHELOR'S THESIS

Author: Mikael Norrbäck
Degree Programme: Information Technology
Supervisor: Kaj Wikman

Title: *Assistance application for x-ray students*

Date: 29.5.2012 Number of pages 24 Appendices 0

Summary

The thesis was done at the request of the degree programme in radiography and radiotherapy at Novia University of Applied Sciences. The degree programme wanted an application that could be of assistance when teaching how to take x-rays. The work consisted of planning and creating an application with the material that was provided by the employer. The application is coded in C# as a WPF project and contains a virtual x-ray process and an information center for the students.

Language: Swedish Key words: virtual x-ray , C#, WPF

OPINNÄYTETYÖ

Tekijä: Mikael Norrbäck
Koulutusohjelma ja paikkakunta: Tietotekniikka, Vaasa
Ohjaaja: Kaj Wikman

Nimike: *Apuohjelma röntgenopiskelijoille*

Päivämäärä 29.5.2012 Sivumäärä 24 Liitteet 0

Tiivistelmä

Opinnäytetyö tehtiin Novian koulutusohjelman Radiografi och strålbehandling pyynnöstä. Opetuksen helpottamiseksi toivottiin tukiohjelma, jota voitaisiin käyttää apuvälineenä opetettaessa röntgenkuvien ottaminen. Työhön kuului suunnitella ja luoda ohjelma työnantajan antamasta materiaalista. Ohjelma on koodattu C#:ssa WPF-projektina ja se sisältää virtuaaliröntgenin ja tietواسeman opiskelijoille.

Kieli: ruotsi Avainsanat: virtuaaliröntgen, C#, WPF

Innehållsförteckning

Ordförklaringar	1
1 INLEDNING	1
1.1 Beställaren	1
1.2 Bakgrund	1
2 PROGRAMMERINGSSPRÅK	2
2.1 Qt	2
2.2 Java.....	3
2.3 C#	3
3 PROGRAM.....	4
3.1 Microsoft Visual Studio	4
3.2 Microsoft Expression Blend	4
3.3 Adobe Photoshop	5
3.4 Notepad++	5
4 TEKNIK.....	6
4.1 Vilket språk har använts?.....	6
4.2 WPF och XAML.....	6
4.3 Användningen av Microsoft Expression Blend.....	7
5 UTFÖRANDE	8
5.1 Programmetts uppbyggnad	8
5.2 Virtuella röntgen	9
5.3 Information	14
6 RESULTAT, DISKUSSION OCH REFLEKTIONER	16
6.1 Reflektioner	16
6.2 Resultat	16
7 KÄLLFÖRTECKNING	17

Ordförklaringar

XAML	eXtensible Application Markup Language.
WPF	Windows Presentations Foundation.
Photoshop	Ett bildredigeringsprogram utvecklat av Adobe.
PSD	Photoshop Document - Ett normalt photoshop dokument.
PSB	Photoshop Big - Ett större photoshop dokument med bättre kvalitet.
Java	Ett programmeringsspråk utvecklat av James Gosling.
Qt	Ett programmeringsspråk.
GNU	En license, de regler som gäller för ett projekt.
C	Ett gammalt programmeringsspråk utvecklat i slutet av -60 och början av -70-talet.
C#	Ett objektorienterat programmeringsspråk till .NET platformen.
C++	Ett objektorienterat programmeringsspråk influerat av C.
PHP	Ett skriftspråk som tex används på webbservers.
Script	Ett litet program som används för att underlätta för användaren. Man kan tex starta flera processer samtidigt med ett skript.
Batch	Samma beskrivning som för script men batchfiler behöver inte ha en användare till hjälp utan datorn kan använda dem själv.

1 INLEDNING

1.1 Beställaren

Beställaren för denna produkt och examensarbete är utbildningsprogrammet för radiografi och strålbehandling vid Yrkeshögskolan Novia. Utbildningsprogrammet finns vid enheten för social- och hälsovård som finns i stadsdelen Roparnäs i Vasa och är till för personer som vill fokusera på social- och hälsovårdsutbildning. Det finns ett behov av ett hjälpmedel vid utläringen av nativbildstagning på grund av att de lokaler samt den apparatur som behövs inte alltid finns tillgänglig. Orsaken till det är att den inte ägs av skolan. Syftet med detta examensarbete är att skapa ett program som ska underlätta för de studeranden som vill lära sig processen kring nativbildstagning.

1.2 Bakgrund

Det material som har använts i detta examensarbete har samlats in av två röntgenskötarstuderanden från Yrkeshögskolan Novia. De har tagit bilder av de vanligaste projektionerna samt beskrivit dem. Till detta har de också givit röntgenbilder av de vanligaste objekten som ska användas i programmet. Av dessa bilder och texter skulle ett hjälpprogram konstrueras som skulle kunna vara till hjälp för studeranden vid utbildningsprogrammet. Kraven för programmet var:

- En process för nativbildstagning.
- Information om röntgens olika delområden.

Utöver det har det varit programmerarens uppgift att designa programmet och att få fram en produkt för beställarens behov.

2 PROGRAMMERINGSSPRÅK

I detta kapitel förklaras det programmeringsspråk som använts i det utförda arbetet samt två närliggande språk som alternativt hade kunnat användas i examensarbetet. Alla tre rubrikerna innehåller ett kodexempel, så man kan se skillnaden i språken.

2.1 Qt

Efter att ha varit intresserade av C++:s utveckling började Haavard Nord och Eirik Chambe-Eng att utveckla ett eget programmeringsspråk, nämligen Qt. År 1995 lanserades Qt av företaget Trolltech och 2008 skaffade Nokia rättigheterna till QT och bytte först namn till QT Software och senare till QT Development Frameworks. /10/

I bakgrunden av Qt-kod körs vanlig C++-kod, i och med detta kan man också blanda in C++-kod om man vill. Utvecklingen av Qt skiljer sig en aning från utvecklingen av andra programmeringsspråk. När utvecklarna stötte på ett problem så försökte de inte hitta den snabbaste eller lättaste vägen för att komma vidare, i stället försökte man hitta den rätta lösningen. *Figur 1* nedan visar ett litet kodexempel i Qt. /2/

```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QPushButton hellobutton("Hello World");
    hellobutton.resize(80, 15);
    hellobutton.show();
    return app.exec();
}
```

Figur 1. Ett "Hello world" exempel kodat i Qt

2.2 Java

Programmeringsspråket Java utvecklades till först av kanadensaren James Gosling. Stora delar av Javas syntax härstammar från C och C++, men är utformad på ett mer lättförståeligt sätt för att underlätta kodandet. En av de väldigt få nackdelar med Java är att det kan upplevas som långsammare och minnesslukande på grund av att allt har gjorts så enkelt som möjligt och därav behöver mer minne för att allt ska fungera. Java är väldigt lättanvänt och passar perfekt till en person som vill börja lära sig programmering. Ett litet kodexempel finns nedan i *Figur 2.* /5/ /6/

```
class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World");
    }
}
```

Figur 2. Kod till "Hello world" i Java

2.3 C#

På uppdrag av Microsoft började Anders Hejlsberg år 1999 att utveckla ett objektorienterat programmeringsspråk för .NET-miljön, nämligen C#. Språket påminner på många sätt om Java. Javas grundare Gosling har också påpekat att C# är en Java-klon, medan Hejlsberg argumenterar att C# i grund och botten är C++. Java och C# liknar varandra till utseendet och har därför också liknande för och nackdelar. C# är ett nytt språk i jämförelse med till exempel C++. Eftersom C# är ett relativt nytt språk så har man tagit fördelarna från andra språk och skapat ett eget språk. På samma sätt som troligtvis andra utvecklare också har gjort. I och med det är C# är väldigt kraftfullt språk. Ett kodexempel i C# finns nedan i *Figur 3.* /3/ /4/

```

class HelloWorld
{
    public static void Main()
    {
        Console.WriteLine("Hello World");
    }
}

```

Figur 3. "Hello world" skrivet i C#.

3 PROGRAM

Här listas de program som använts för att uppnå resultatet.

3.1 Microsoft Visual Studio

Microsoft Visual Studio är en utvecklingsmiljö av Microsoft som inriktat sig på applikationer och webapplikationer för .NET-miljön. Studio har stöd för väldigt många programmeringsspråk som t.ex. C#, C/C++ och många fler. Vill man använda sig av mindre kända språk så kan man installera ett flertal tjänster för att kunna programmera i det önskade språket. Det finns flera olika utgåvor av Microsoft Visual Studio. Express, Premium, Professional och Ultimate. De två förstnämnda är mer inriktade till den vardagliga hobbyprogrammeraren även om de innehåller funktioner för mer än det. Professional och Ultimate är för de som jobbar dagligen och utvecklar produkter. De innehåller funktioner för diverse analyser och tester med mera. /11/ /8/

3.2 Microsoft Expression Blend

Microsoft Expression Blend är ett verktyg för att skapa eller redigera den grafiska delen av applikationer man gjort med till exempel Microsoft Visual Studio. Vill man snygga till de applikationer man skapat, går det enkelt att öppna enskilda filer eller hela projekt med blendern för att snygga till utseendet. Blend är baserat på XAML-kod så det går väldigt enkelt att använda programmet vid sidan om Studio när man jobbar med WPF-projekt. Man kan även testköra sina projekt direkt eftersom programmet innehåller en kompilator. /7/

3.3 Adobe Photoshop

Adobe Photoshop är ett bildredigeringsprogram som har väldigt många funktioner för att redigera bilder av olika slag. Man kan göra enkla saker som att t.ex. ändra färger på en bild eller mera avancerade saker som att t.ex. redigera ihop flera bilder till ett projekt. Många kända professionella program stödjer Photoshops filändelser PSD och PSB eftersom Photoshop är så pass stort. Det går också att spara sina projekt i .gif- eller .jpeg-format så att program som inte stödjer Photoshops filändelser också kan öppna projekten. Risken med detta är att kvaliteten på projekten blir mer eller mindre sämre. Photoshop finns tillgängligt i tiotals olika språk för att nå ut till så många eventuella användare som möjligt. /1/

3.4 Notepad++

Notepad++ är ett gratisprogram under GNU- licensen som kan användas till att redigera källkod. Notepad++ en mer avancerad version av Notepad som finns med som standard i Windows. Programmet har stöd för tiotals olika språk så som C++ och PHP samt olika script- och batch-kommandon. Känner man att man behöver jobba med flera dokument samtidigt så är det inga problem eftersom Notepad++ använder sig av flikar så man kan öppna flera filer samtidigt. Om programmet i sig inte är tillräckligt finns det flera olika tillägg att installera som kan göra användningen mer bekväm, till exempel en stavningskontroller eller möjligtvis en extra clipboard om man inte nöjer sig med extra flikar. /9/

4 TEKNIK

4.1 Vilket språk har använts?

Valet av programmeringsspråk berodde på ett par orsaker. Det hade varit möjligt att skriva den praktiska delen av examensarbetet i alla tre språken och kanske också fler, men valet blev C# av följande orsaker.

- Mest erfarenhet i C#.
- Goda framtidsutsikter.

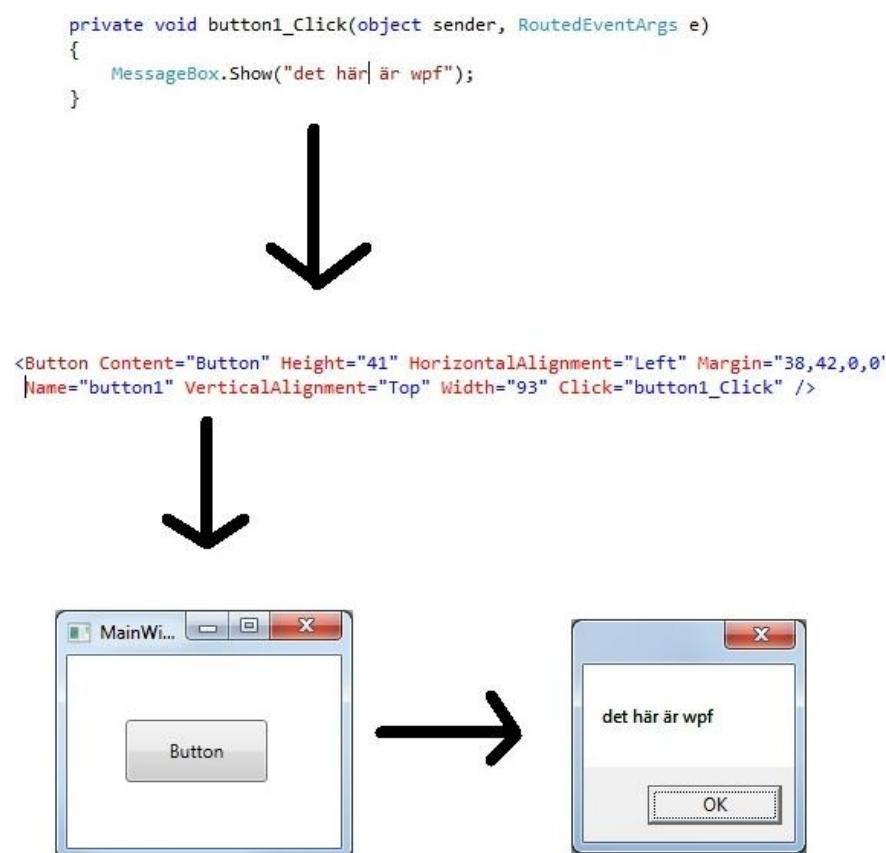
Först och främst så känns det som att man har använt C# mest inom utbildningen, även om man fått grunderna i de andra språken också. En annan orsak är att det känns som att C# är det enda av dessa som verkar riktigt tilltalande inför framtiden, speciellt WPF. WPF som användes i examensarbetet är intressant på grund av de möjligheter man har gällande den visuella delen av ett programmet, det finns väldigt många verktyg och funktioner för att få till ett snyggt resultat. Alla visuella objekt kan modifieras enskilt med XAML-kod. Alla funktioner med mera är kodade i C# medan alla visuella objekt är i XAML-format och som i sin tur är ihopkopplade med diverse skapade funktioner för att bilda programmets helhet. /12/

4.2 WPF och XAML

WPF är nästa generations Windows Forms. Stilen har varit tillgänglig i Microsoft Visual Studio sedan 2005 ävenom man var tvungen att skriva allt manuellt. När Microsoft lanserade Visual Studio 2008 så hade man implementerat "drag and drop-funktioner" för alla visuella objekt tillgängligt i WPF. WPF är i princip två olika programmeringsspråk. Man använder sig av C# för att skapa klassiska klasser och funktioner, för att sedan använda visuella fönster skapade i XAML-kod för att

kunna visa sina skapelser utåt. /8/

XAML är ett språk som till namnet och utseendet liknar HTML. XAML:s primära uppgift är att skapa det visuella för ett WPF-projekt samt definiera parametrar för till exempel knappar, paneler med mera. Som *Figur 4* nedan visar så används XAML som en mellannivå för att hålla samman C#-koden med det visuella som användaren av den färdiga applikationen ser. /12/



Figur 4. En bild som visar hur XAML sammanknyter funktioner med användargränssnittet.

4.3 Användningen av Microsoft Expression Blend

När man känner att det är dags att börja tänka över utseendet på sitt skapade projekt så kan man faktiskt komma väldigt långt med att endast använda Microsoft Visual Studio, för att snygga till det visuella och få det hela att se mer professionellt ut. Ändra storlekar på objekt samt

avstånd dem emellan går bra, tills man märker att allt för många objekt sitter fast i grundfönstret eller i någon grid. Det är fullt möjligt att göra allt detta i Visual Studio, men när man märker att man kan göra samma sak på ett par sekunder med Microsoft Expression Blend så blir man lite snopen över all den tid man lagt ner på att snygga till saker i Visual Studio.

5 UTFÖRANDE

5.1 Programmets uppbyggnad

Programmet är uppbyggt av två delar.

- En virtuell röntgen.
- Ett informationscente.

I den virtuella röntgen kan man ställa in röntgenvärden för en utvald kroppsdel, för att sedan se resultatet i form av en röntgenbild som varierar beroende på valda värden. Den andra delen är ett litet informationscenter uppbyggt av flikar, en flik för varje område. Under ett besök till röntgenavdelningen vid Vasa Centralsjukhus så fick man se hur programmen fungerade, så att de som använder detta program skall kunna övergå så smidigt som möjligt från virtuella övningar till riktig praktik. De båda delarna är sammanknutna med varandra med hjälp av ett huvudfönster som också är det fönster som öppnas först när man startar programmet. Huvudfönstret är gjort på enklast möjliga sätt, tre knappar av vilket två stycken går till var och en av delarna, samt en tredje som helt enkelt stänger av programmet. I *Figur 5* nedan kan man se hur huvudfönstret ser ut.



Figur 5. En bild av programmets startsida.

Tekniken bakom startsidan är inte speciellt anmärkningsvärd, två knappar som tar användaren vidare till önskad del av programmet. Det som kan nämnas är att texterna på knapparna är gjorda i ett textblock som i sin tur sitter i en stackpanel. Det är inte på något sätt nödvändigt, med det ger en möjligheten att enklare designa utseendet på knapparna genom att t.ex. lägga till en bild under stackpanelen. I XAML-koden ser det ut på följande sätt som *Figur 6* visar.

```
<Button>
  <StackPanel>
    <TextBlock/>
  </StackPanel>
</Button>
```

Figur 6 XAML-kod som visar tekniken bakom knapparna.

5.2 Virtuella röntgen

Med virtuell röntgen får man röntga valfri kroppsdel genom att välja en kroppsdel och ställa in röntgenvärden, för att sedan se ett varierande resultat beroende på vilka värden man ställt in.

Först möts man av ett fönster där man skall välja en kroppsdel genom att

kryssa i valfri kroppsdel i en lista. När man valt en kroppsdel så måste man ännu välja om man vill göra en normal eller en lateral röntgen, dvs. röntgen ovanifrån eller från sidan. Detta gör man genom att kryssa i ett alternativ. Man kan inte komma vidare till nästa fönster om man inte har valt en kroppsdel att ställa in värden för. I *Figur 7* kan man se hur valfönstret ser ut.



Figur 7. En bild som visar hur valfönstret ser ut.

Valfönstret var lite knepigt att koda till en början. Det finns ett dussintal ikryssningsmöjligheter och endast en av dem får vara ikryssad när man går vidare. Går säkert att göra på flera sätt, men i detta fall valdes en lösning innehållande två funktioner. En funktion som kontrollerar om någon box är ikryssad samt en funktion som ser till att endast en box kan vara ikryssad. Om man väljer ett annat alternativ efter det första avkryssas den första automatiskt. Knappen som tar en vidare från valfönstret innehåller en lång rad if-satser för varje alternativ, så att man kommer vidare till rätt fönster. I *Figur 8* kan man se hur funktionen som ser till att endast en box kan vara ikryssad ser ut.

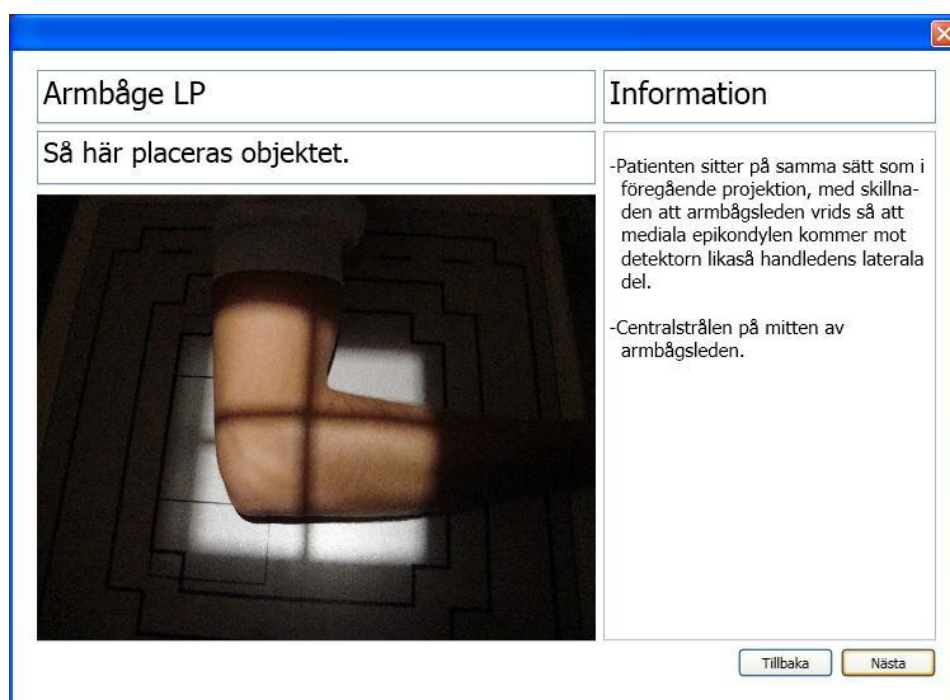

```

private void onlyOneIsChecked(CheckBox valueIsChecked)
{
    foreach (TreeViewItem i in treeView.Items)
    {
        foreach (CheckBox j in i.Items)
        {
            if (j != valueIsChecked)
            {
                j.IsChecked = false;
            }
        }
    }
}

```

Figur 8. Funktionen som ser till att endast en box kan vara ikryssad.

När man har valt en kroppsdel kommer man vidare till nästa fönster som i princip bara har en funktion. Vilken kroppsdel man än väljer så kommer detta fönster att visa upp ett exempel på hur man bör placera patienten på röntgenbordet, för att få en så bra bild som möjligt samt direktiv på hur man bör installera patienten för att få en så bra bild som möjligt. Hur det ser ut kan man se i *Figur 9* Nedan.

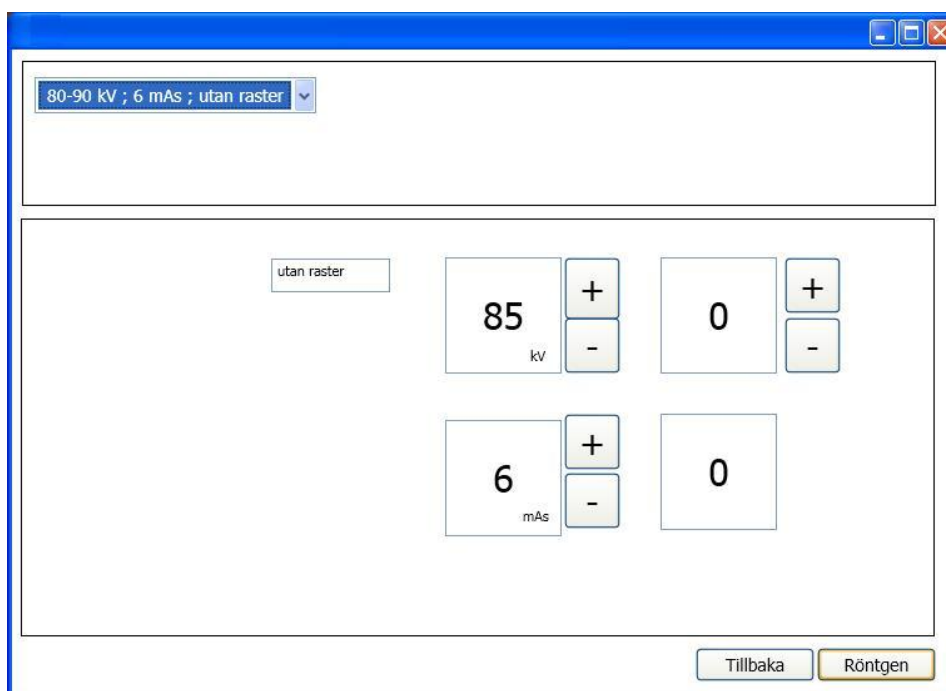


Figur 9. En bild av fönstret som ger information om det valda alternativet.

Konstruktionen av informationsfönstret var inte speciellt svårt. Ett par textboxar och textblock behövdes för att få fram informationen som behövs. De är i sin tur inramade med rektangelramar för att få det hela

att se lite snyggare ut. Knapparna är det enda som är skrivet med C#, eftersom de är funktioner, resten är gjort med XAML-kod.

Nästa fönster som kommer fram när man klickar sig vidare är platsen där man ställer in de olika värdena i den virtuella röntgenmaskinen. Uppe till vänster finns en dropdown-box där man måste välja ett alternativ. Varje alternativ innehåller en del olika värden som fylls i. Meningen med detta är att de som använder programmet ska med sin egen kunskap ta reda på vilket alternativ som är mest passande för den kroppsdel man har valt. Det går att ställa in värden manuellt, men man kan inte komma vidare till nästa sida om man inte valt ett alternativ först. I *Figur 10* ser man hur fönstret ser ut.



Figur 10. Fönstret där man ställer in röntgenvärden.

Bilden av hur fönstret där man ställer in värden skulle se ut var ganska klar redan från början. Hur allt skulle se ut och fungera i bakgrunden samt hur kommunikationen med nästa fönster skulle gå till, var däremot lite oklart. Nästa fönster innehåller en textbox som visar de värden som var inställda innan man klickar vidare. Själva röntgenbilden i det sista fönstret skall också variera beroende på det alternativ man valt i comboboxen uppe till vänster. Detta löstes med de två variablerna

currentValues och chosenValues. CurrentValues lagrar de värden som man kan ställa in manuellt och currentValues lagrar det alternativ som väljs i comboboxen. De skickas sedan vidare till nästa fönster i funktionen som startas av knappen Röntgen. Hur funktionen ser ut ser man i *Figur 11*.

```
public void BTN_Röntgen_Click(object sender, RoutedEventArgs e)
{
    if (ComboBoxValues.SelectedValue == null)
    {
        MessageBox.Show("Du måste välja ett alternativ ur listan innan du fortsätter.", null, MessageBoxButton.OK);
    }
    else
    {
        string currentValues;
        currentValues = txtkV.Text.ToString() + " , " +
            txtmAs.Text.ToString() + " , " +
            txtRaster.Text.ToString();
        var newtestfönsterBILD =
            new ArmbågeLPBild(currentValues, chosenValues);
        newtestfönsterBILD.Show();
        this.Close();
    }
}
```

Figur11. Funktionen som sänder vidare två variabler samt öppnar nästa fönster.

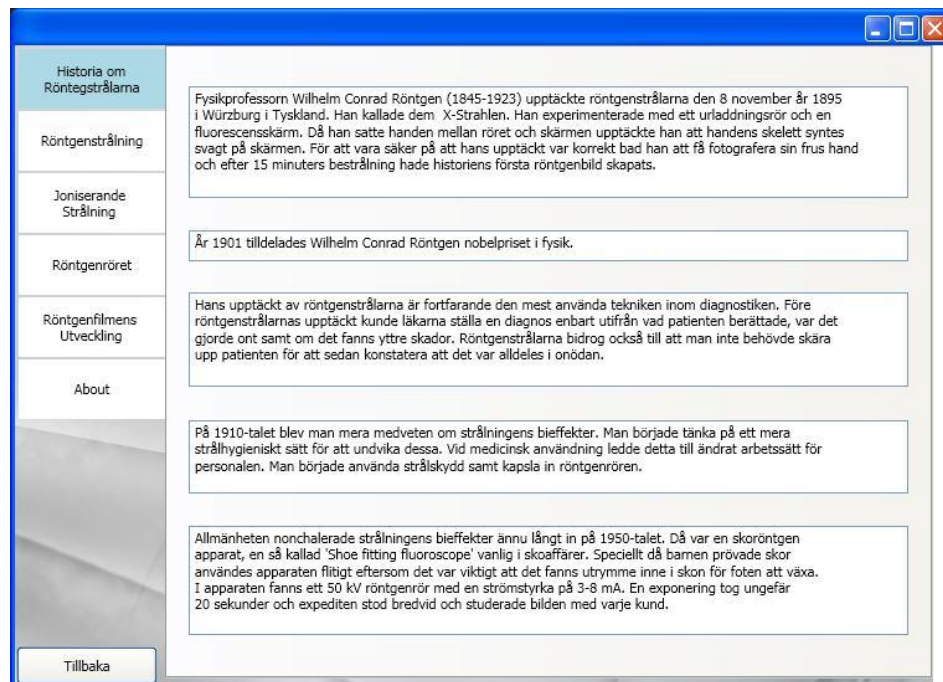
I det sista fönstret får man se ett resultat i form av en bild som varierar lite grann beroende på inställningarna i föregående fönster. Den rätta bilden väljs genom värden som variabeln chosenValues har lagrat undan. Har man för låga värden så blir bilden mörk och har man för höga värden blir bilden för ljus. De inställningar man har använt kan man se i en ruta längst ner i fönstret. *Figur 12* visar hur det ser ut.



Figur 12. En bild av det sista fönstret i den virtuella röntgenprocessen.

5.3 Information

Den andra delen av programmet är ett litet upplagsverk med information kring röntgen, t.ex. strålningstyper. Denna del av programmet är bara ett fönster, men som i sin tur har en del flikar med text och bilder av till exempel röntgenröret, röntgens historia samt modern röntgen. Här finns också en "about-flik" längst ner, där man kan se vilka som varit delaktiga i programutvecklingen samt kontakuppgifter. Hela fönstret är i princip kodat med XAML-kod. Fliken som är aktiv byter färg till ljusblå. Inte direkt något nödvändigt, men det var något som passade bra in så man har koll på vilken flik man befinner sig i. En bild av den slutgiltiga versionen av fönstret kan ses i *Figur 13*.



Figur 13. En bild av hur informationsdelen ser ut.

6 RESULTAT, DISKUSSION OCH REFLEKTIONER

6.1 Reflektioner

Känns som att detta lärdomsprov var väldigt givande och framför allt lärorikt. Det finns många saker jag kommer ta med mig in i arbetslivet. Till att börja med så borde jag ha spenderat mer tid med att planera själva projektet. Tiden användes tvärtemot hur man egentligen borde ha gjort, så nästan gång jag tar mig an ett projekt kommer jag använda mycket mer tid för att planera så noga som möjligt för att underlätta på alla plan. Jag var lite för ivrig i början för att få se ett snabbt resultat. En annan närliggande sak som slog mig under slutet av projektet, var att om det är lätt i början så blir det svårare mot projektets slut. Rent praktiskt så är en stor del av programmet hårdkodat, vilket inte alltid är det bästa sättet att koda på, speciellt när man har funktioner som har ett tiotal med valmöjligheter. Det hade varit mycket enklare att ha en textfil eller en XML-fil med data då hade man inte behövt koda lika mycket samt att det hade varit lättare att redigera information i framtiden. Även folk som inte besitter någon kunskap inom programmering hade enkelt kunnat lägga till information i programmet.

6.2 Resultat

Överlag är jag rätt nöjd med hur examensarbetet artade sig mot slutet. Utseendemässigt så blev slutprodukten i stil med vad jag tänkte när jag börja med det sommaren 2011. En del funktioner har tagits bort och ett par funktioner har lagts till, men överlag blev det ungefär som jag tänkte mig. Det finns alltid möjlighet att gå in och ändra kod om man har kunskap till det, snygga till saker och ting går också om man känner för det. Har man fler saker man skulle vilja lägga till i programmet förutom informationsdelen och den virtuella röntgen, så borde det gå att göra utan några större problem. Tror dock att programmet fyller sin uppgift mer än väl, som det är nu.

7 KÄLLFÖRTECKNING

- /1/ Adobe Photoshop
 http://en.wikipedia.org/wiki/Adobe_Photoshop
 (läst: 20.1.2012)
- /2/ C++ GUI Programming with Qt 3
 av Jasmin Blanchette, Mark Summerfield
 (läst: 9.4.2012)
- /3/ C#
 http://en.wikipedia.org/wiki/C_Sharp
 (läst: 10.1.2012)
- /4/ C# The Basics
 av Vijay Mukhi, Sandeep Shanbhag, Sonal Mukhi
 (läst: 9.4.2012)
- /5/ Java
 [http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))
 (läst: 10.1.2012)
- /6/ Java
 <http://www.webdotdev.com/nvd/articles-reviews/java/java-advantages-and-disadvantages-1042-1.html>
 (läst: 9.4.2012)
- /7/ Microsoft Expression Blend 2 for Windows
 av Corey Schuman, Robert Reinhardt
 (läst: 1.2.2012)
- /8/ Microsoft Visual Studio
 http://en.wikipedia.org/wiki/Microsoft_Visual_Studio
 (läst: 12.1.2012)
- /9/ Notepad++
 <http://en.wikipedia.org/wiki/Notepad%2B%2B>
 (läst: 13.1.2012)
- /10/ Qt
 [http://en.wikipedia.org/wiki/Qt_\(framework\)](http://en.wikipedia.org/wiki/Qt_(framework))
 (läst: 10.1.2012)

- /11/ Visual Studio 2010 All-In-One for Dummies
av Andrew Moore
(läst: 9.4.2012)
- /12/ WPF in C# 2008: Windows Presentation Foundation .NET 3.5
av Matthew MacDonald
(läst: 9.4.2012)